# Phase Transitions in Optimization Problems

Alexander K. Hartmann

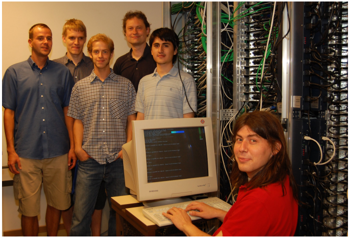
Institute of Physics
University of Oldenburg

DPG Physics School, Bad Honnef, 14. September 2012

# Research Group Computational Physics

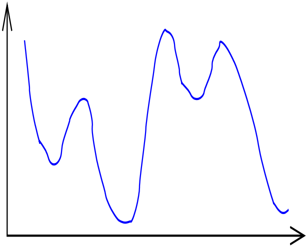
"Complex behavior of discrete systems
in Physics, Biology, Mathematats and Computer Science"

Computer Simulations
New algorithms



few group members

Optimization algorithms
Development/application

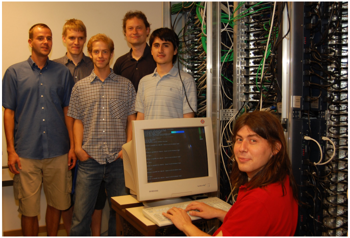


# Research Group Computational Physics

“Complex behavior of discrete systems
in Physics, Biology, Mathematcs and Computer Science”

## Computer Simulations

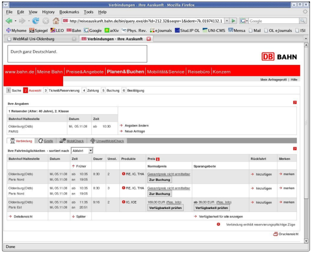
New algorithms



few group members

## Optimization algorithms

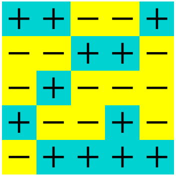
Development/application




MANY variables

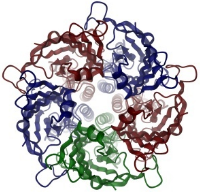
## Disordered magnets

alloys, e.g., iron/gold
spin glasses
random-field systems



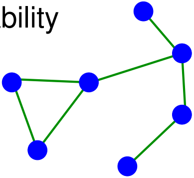
## Biologie

RNA secondary structures
comparison of proteins
bats



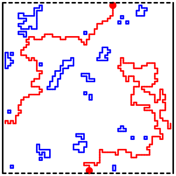
## Phasen transitions in optimization problems

Vertex Cover
Satisfiability



## Percolation problems

systems carrying information



# Museums

London

Zürich

Paris

Are they safe?

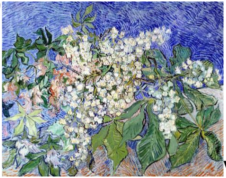
# Museums

London

Zürich

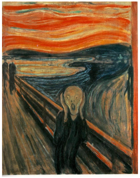
Paris

Are they safe?



van Gogh

Cezanne



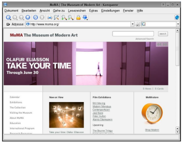
Munch

# Saftey Measures

Access

Copies

Virtual museums

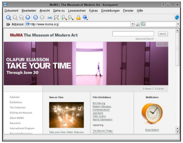


# Saftey Measures

Access

Copies

Virtual museums




**Guards:**

hyper active

efficient (lazy)

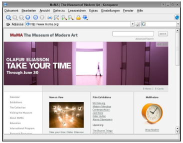
unefficient

# Saftey Measures

Access

Copies

Virtual museums




**Guards:**

hyper active

efficient (lazy)

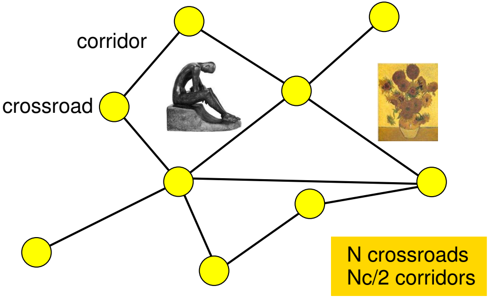
unefficient

**Modelling:**

- Reduction
- Mathematical description
- Solution

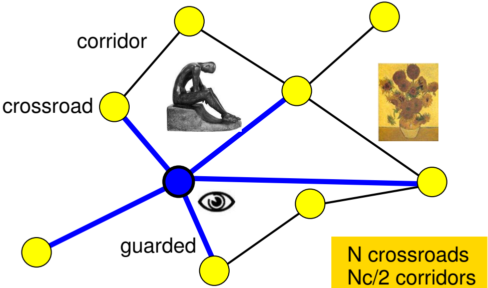
# Vertex-Cover Problem

- Prototypical problem of theoretical Computer Science
- Model



# Vertex-Cover Problem

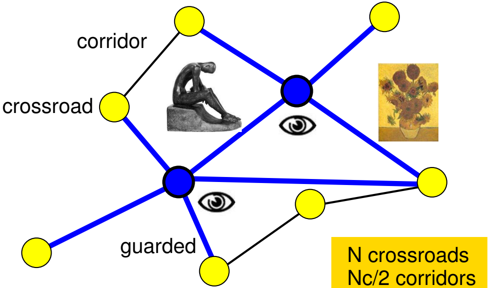
- Prototypical problem of theoretical Computer Science
- Model




$X = xN$ guards
guard only adjacent corridors

# Vertex-Cover Problem

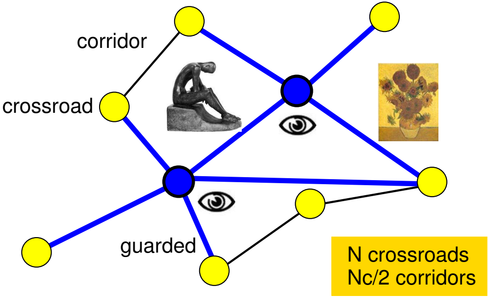
- Prototypical problem of theoretical Computer Science
- Model




$X = xN$ guards
guard only adjacent corridors

# Vertex-Cover Problem

- Prototypical problem of theoretical Computer Science
- Model




$X = xN$ guards
guard only adjacent corridors

- Optimization probl. A: minimize number of guards
  Optimization probl. B: minimize number of unguarded corr.
- Mathematically: museum = graph
  Vertex-cover problem = NP-complete

# Heuristics

- given : museum (graph) $G = (V, E)$
- wanted: minimum number of guards (problem (A))
- **algorithm** `min_cover`(G)

**begin**

$V' = \emptyset$

**while**(there are unguarded corridors)

**do**

select crossroad $i \in V$ with highest degree $d_i$

guard crossroad: $V' = V' \cup \{i\}$

remove edged adjacent to $i$ from $E$

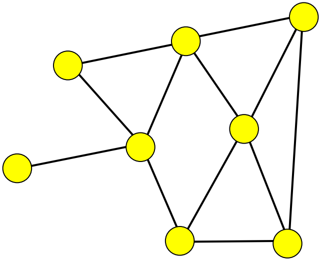
**end**

**return** $V'$

**end**

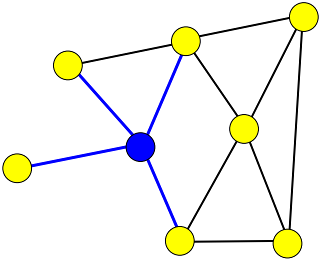
# Leaf-removal Algorithm

- Where to put guards without violating minimum condition?



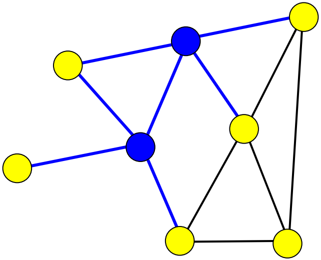
# Leaf-removal Algorithm

- Where to put guards without violating minimum condition?



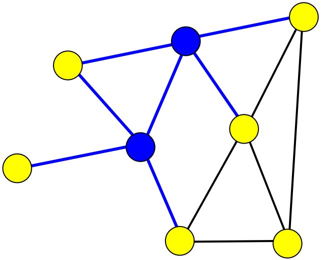
# Leaf-removal Algorithm

- Where to put guards without violating minimum condition?



# Leaf-removal Algorithm

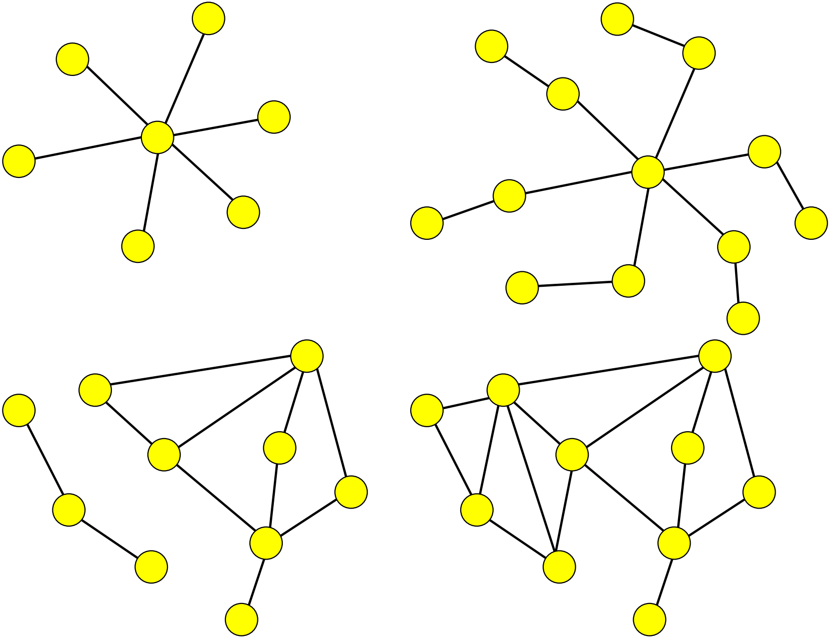
- Where to put guards without violating minimum condition?



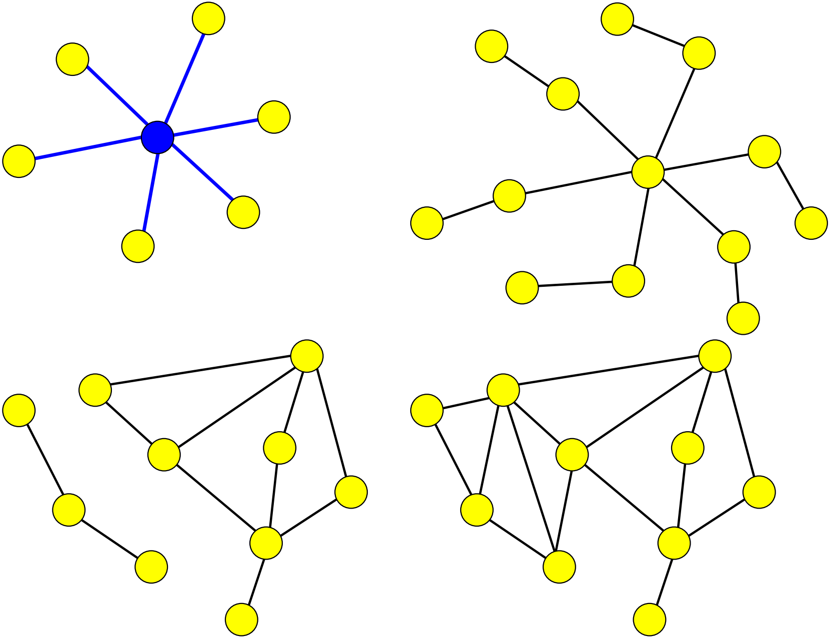
- **algorithm** `leaf_romoval`(G)
  **begin**
      **while**(there are leaves)
          guard neighbor *i* of leaf and remove adjacent edges
  **end**

- Remaining graph is called core $\rightarrow$ exact algorithm

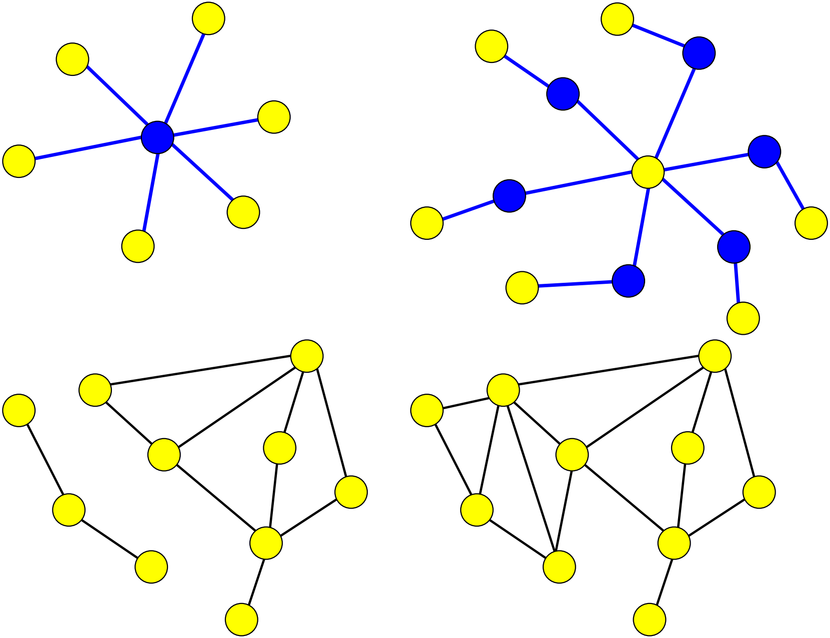
Solve min. problem (A) for handout examples **NOW!**



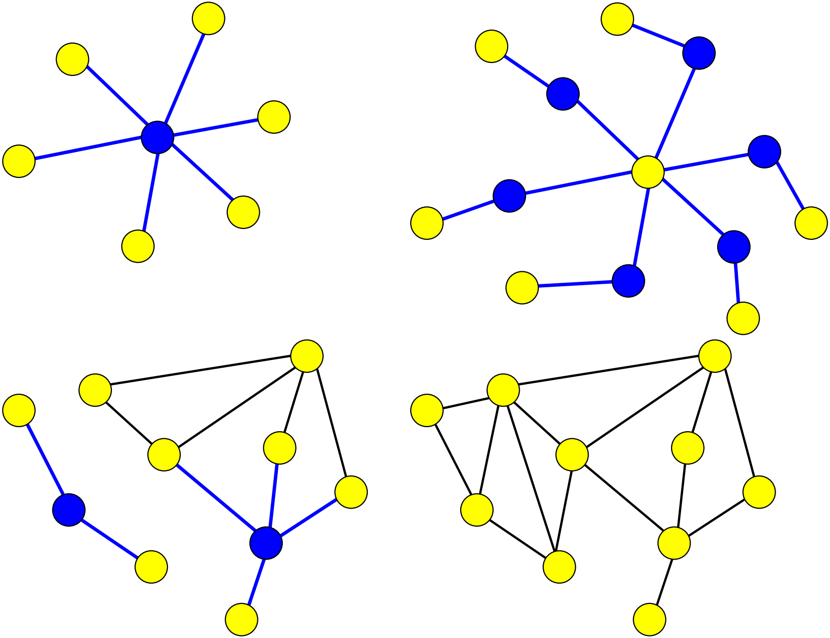
Solve min. problem (A) for handout examples **NOW!**



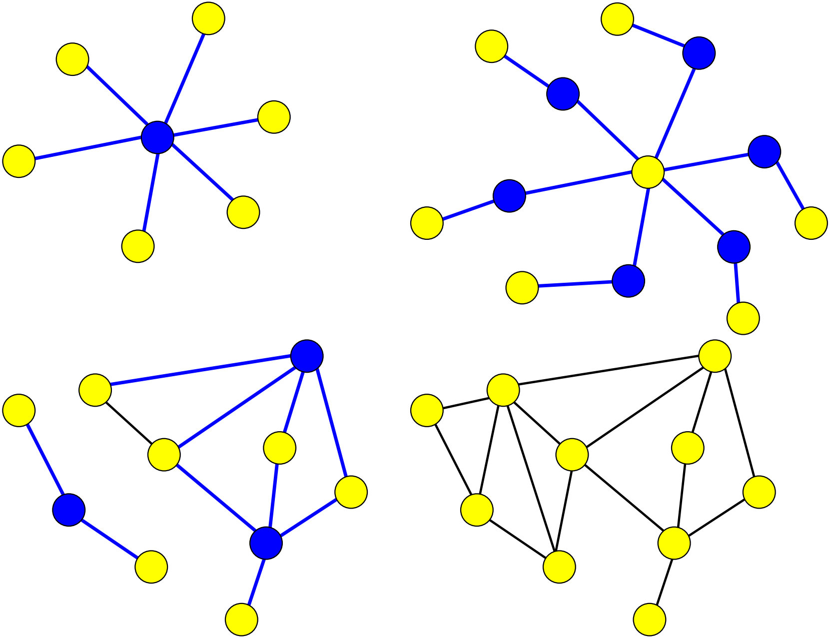
Solve min. problem (A) for handout examples **NOW!**



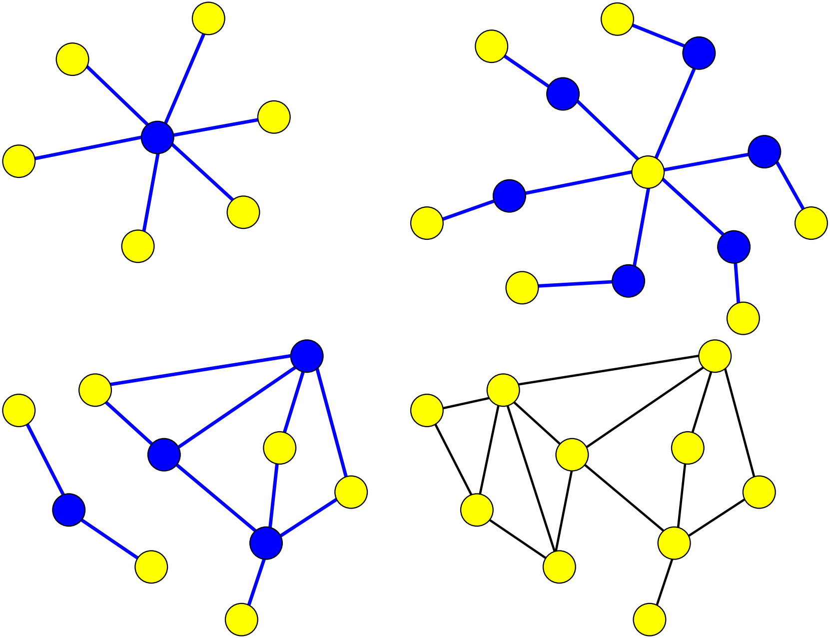
Solve min. problem (A) for handout examples **NOW!**



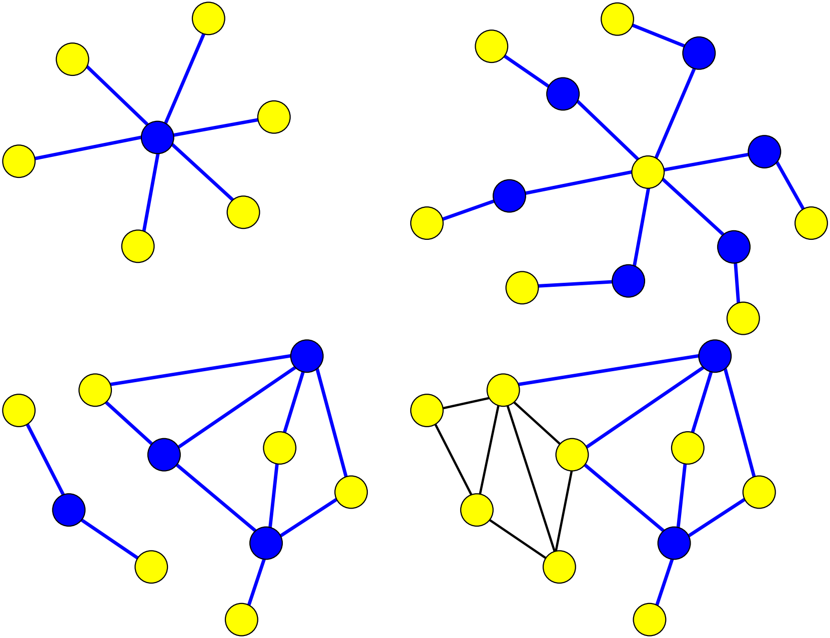
## Solve min. problem (A) for handout examples **NOW!**



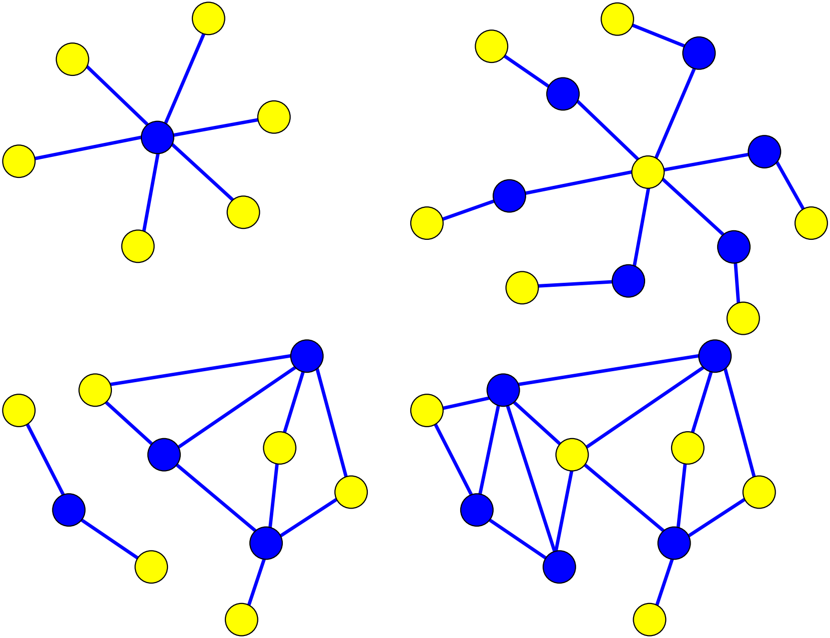
Solve min. problem (A) for handout examples **NOW!**



Solve min. problem (A) for handout examples **NOW!**



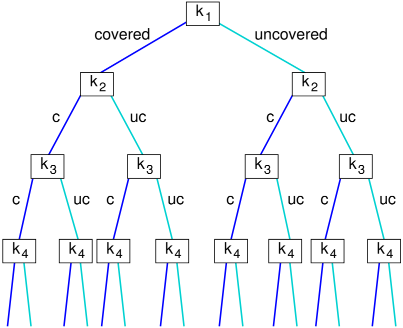
Solve min. problem (A) for handout examples **NOW!**



# Branch-and-bound Algorithm

Task: min. # of uncov. edges (probl. B)

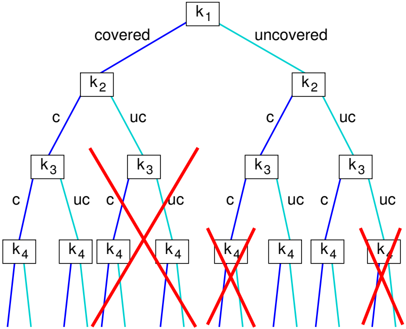
Complete algorithm:
(basically) enumerate all states



# Branch-and-bound Algorithm

Task: min. # of uncov. edges (probl. B)

Complete algorithm:
(basically) enumerate all states




Avoid subtrees w/o solutions

*best* $=$ minimum so far

$X'$ $=$ # of curr. covered vertices

$\Rightarrow$ cover $F := X - X'$ vertices

List $F$ vertices with highest current degrees. Ex. ($F = 3$):

$n_1$: 5 edges
$n_2$: 3 edges
$n_3$: 3 edges
$n_4$: 2 edges
$n_5$: 2 edges
...

$d_{\max} \equiv \sum_{i=1}^{F} d(n_i)$

If (#(uncovered edges)$-d_{\max}$ >best) $\rightarrow$ bound!

# Random Objects: Lego Buildings

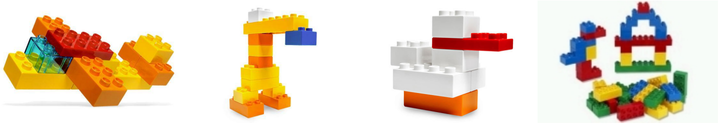
- Basic random experiment:

  throw a dice

- Complex random experiment:

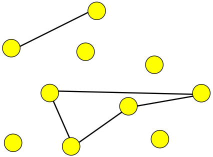
  reach into box with Legos

$\Rightarrow$ random building/object:

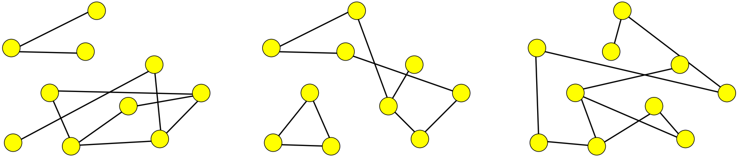


# Random Museums

- Generated in the computer
  $\rightarrow$ defines statistical properties
- $N$ crossroads, $cN/2$ randomly chosen corridors
- $c = 1$



- $c = 2$



- Für each museum: is fraction $x = X/N$ of guards enough?

# Phase Transitions

- Physics, cooperation of many “particles”
- Water: ice $\longleftrightarrow$ fluid $\longleftrightarrow$ vapor


- Street:
  few cars: smooth traffic
  many cars: traffic jam!

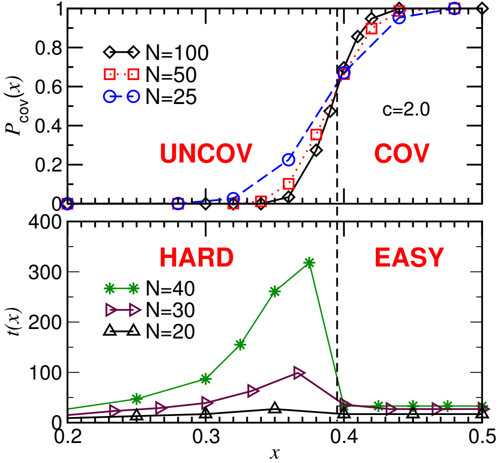

# Phase Transition

- Ensemble: Erdös-Rényi random graphs:
  $N$ vertices and $cN/2$ random edges
- Numerically: averaging over different realizations
- $c = 2$

  Probability to cover

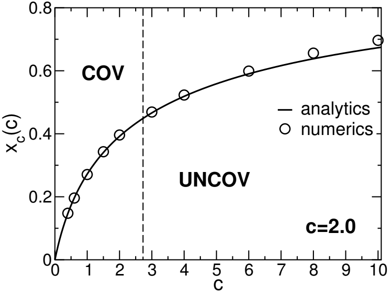
  Running time = number of nodes in branching tree

  [M. Weigt and AKH,
  Phys. Rev. Lett. 2000]



# Phase Diagram

- Finite-size scaling analysis of numerical results: extrapolation ($N \to \infty$) $\Rightarrow$ phase boundary $x_c(c)$
- Analytical treatment: $\Leftrightarrow$ spin-glass or hard-core gas
  Stat. Mech. methods:
  replica trick/cavity approach
  exact for $c \leq e \approx 2.718$
  (replica symmetry = RS)
  $c > e$: RS breaking (RSB)
  [M. Weigt & AKH, PRE 2001]

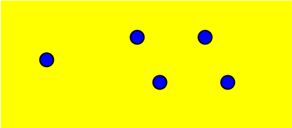



- Can one see cluster structure/RSB numerically ?

# Hierarchical Clustering

- Start: $Z$ configs $=$ $Z$ single configuration clusters $C_j = \{\underline{x}^j\}$ initial distances $d(C_j, C_l) = d_{\mathrm{Hamming}}(\underline{x}^j, \underline{x}^l)$
- Merge iteratively nearest clusters $C_{\mathrm{new}} = C_\alpha \cup C_\beta$, update $d(C_{\mathrm{new}}, C_j)$ $(j \neq \alpha, \beta)$, until one cluster left.

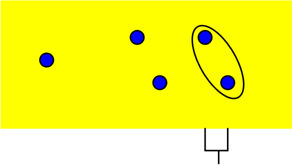
  [J.H. Ward, J. Am. Stat. Assoc. 1963]



# Hierarchical Clustering

- Start: $Z$ configs $=$ $Z$ single configuration clusters $C_j = \{\underline{x}^j\}$ initial distances $d(C_j, C_l) = d_{\text{Hamming}}(\underline{x}^j, \underline{x}^l)$
- Merge iteratively nearest clusters $C_{\text{new}} = C_\alpha \cup C_\beta$, update $d(C_{\text{new}}, C_j)$ $(j \neq \alpha, \beta)$, until one cluster left.

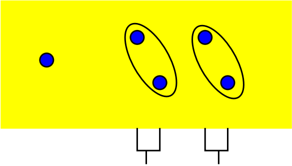
  [J.H. Ward, J. Am. Stat. Assoc. 1963]



# Hierarchical Clustering

- Start: $Z$ configs $= Z$ single configuration clusters $C_j = \{\underline{x}^j\}$
  initial distances $d(C_j, C_l) = d_{\text{Hamming}}(\underline{x}^j, \underline{x}^l)$
- Merge iteratively nearest clusters $C_{\text{new}} = C_\alpha \cup C_\beta$, update $d(C_{\text{new}}, C_j)$ $(j \neq \alpha, \beta)$, until one cluster left.

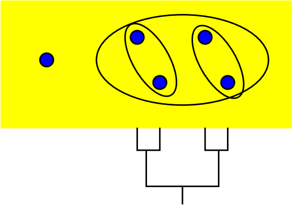
  [J.H. Ward, J. Am. Stat. Assoc. 1963]



# Hierarchical Clustering

- Start: $Z$ configs $=$ $Z$ single configuration clusters $C_j = \{\underline{x}^j\}$ initial distances $d(C_j, C_l) = d_{\text{Hamming}}(\underline{x}^j, \underline{x}^l)$
- Merge iteratively nearest clusters $C_{\text{new}} = C_\alpha \cup C_\beta$, update $d(C_{\text{new}}, C_j)$ $(j \neq \alpha, \beta)$, until one cluster left.

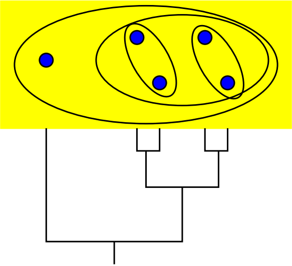
  [J.H. Ward, J. Am. Stat. Assoc. 1963]



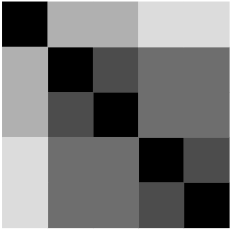
# Hierarchical Clustering

- Start: $Z$ configs $= Z$ single configuration clusters $C_j = \{\underline{x}^j\}$ initial distances $d(C_j, C_l) = d_{\text{Hamming}}(\underline{x}^j, \underline{x}^l)$
- Merge iteratively nearest clusters $C_{\text{new}} = C_\alpha \cup C_\beta$, update $d(C_{\text{new}}, C_j)$ $(j \neq \alpha, \beta)$, until one cluster left.

[J.H. Ward, J. Am. Stat. Assoc. 1963]



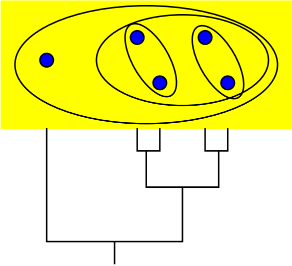
$d(\underline{S}^\alpha; \underline{S}^\beta)$:



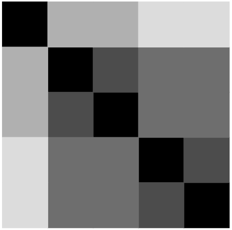
# Hierarchical Clustering

- Start: $Z$ configs $= Z$ single configuration clusters $C_j = \{\underline{x}^j\}$ initial distances $d(C_j, C_l) = d_{\mathrm{Hamming}}(\underline{x}^j, \underline{x}^l)$
- Merge iteratively nearest clusters $C_{\mathrm{new}} = C_\alpha \cup C_\beta$, update $d(C_{\mathrm{new}}, C_j)$ $(j \neq \alpha, \beta)$, until one cluster left.

[J.H. Ward, J. Am. Stat. Assoc. 1963]



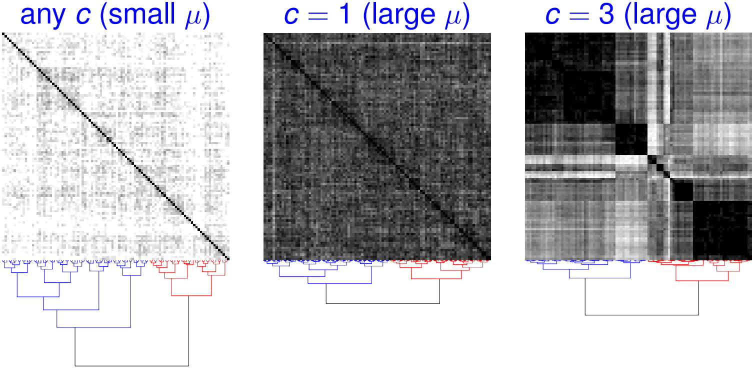
$d(\underline{S}^\alpha, \underline{S}^\beta)$:



- Any set of configs can be clustered $\to$ Does it match? cophenetic correlation: $\mathcal{K} \equiv [d \cdot d_c]_G - [d][d_c]_G$, ($d_c$: distance along tree, $[..]_G$: disorder average) Hubert's $\Gamma$ statistics: correlation $d \leftrightarrow \frac{d_{\max}}{z}$ clustering

# VC: hierarchical clustering

(grand-canonical ensemble (chem. pot. $\mu$) using PT/(MC)$^3$)




[W. Barthel & AKH, PRE 2004]

| | |
|---|---|
| (large $\mu$): | no structure ("paramagnet") |
| $c < e$: | solution cluster has no structure |
| $c > e$: | hierarchy of solution clusters |

cophenetic correlation $K(N)$: decreases/grows for $c < e/c > e$

Complex phase space organization for $c > e$

# Linear Programming (LP)

- B&B algorithm or stochastic methods $\rightarrow$
  move inside configuration space (usually no optimum)
- LP: move outside configurations space (always optimum)
- For each node $i$: variable $x_i \in [0, 1]$:
  $x_i = 1 \leftrightarrow$ covered  $x_i = 0 \leftrightarrow$ uncovered
  $x_i \in ]0, 1[ \leftrightarrow$ undecided
- Each of the $M$ edges $\{j, k\} \rightarrow$ constraint $x_j + x_k \geq 1$
- Objective function: $x \rightarrow$ min

## VC as LP:

**Minimize** $x = \sum_{i=1}^{N} x_i$

**Subject to** $0 \leq x_i \leq 1 \quad \forall\, i \in V$

$x_j + x_k \geq 1 \quad \forall\, \{j, k\} \in E$

Use Simplex algorithm to solve LP

[G.B. Dantzig, Bull. Amer. Math. Soc. 1948] [http://lpsolve.sourceforge.net/5.5/]

# Example

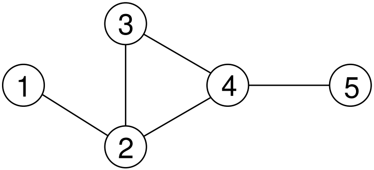



Figure : Example graph with $N = M = 5$

Corresponding LP:

**Minimize** $x = x_1 + x_2 + x_3 + x_4 + x_5$

**Subject to** $0 \leq x_i \leq 1 \quad \forall\, i \in V$

$$
\begin{aligned}
x_1 + x_2 &\geq 1\\
x_2 + x_3 &\geq 1\\
x_2 + x_4 &\geq 1\\
x_3 + x_4 &\geq 1\\
x_4 + x_5 &\geq 1
\end{aligned}
$$

# Example

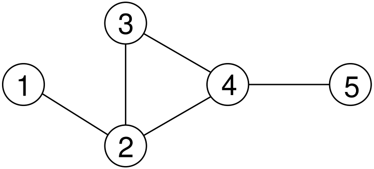



Figure : Example graph with $N = M = 5$

Corresponding LP:

**Minimize** $x = x_1 + x_2 + x_3 + x_4 + x_5$

**Subject to** $0 \leq x_i \leq 1 \quad \forall\, i \in V$

$$x_1 + x_2 \geq 1$$
$$x_2 + x_3 \geq 1$$
$$x_2 + x_4 \geq 1$$
$$x_3 + x_4 \geq 1$$
$$x_4 + x_5 \geq 1$$

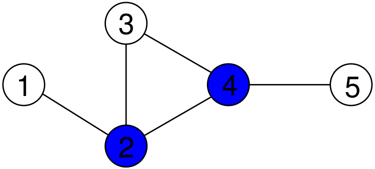
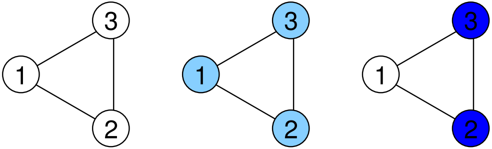



Figure : Minimum VC

**Solution:** $x_1 = 0,$ $x_2 = 1,$ $x_3 = 0,$ $x_4 = 1,$ $x_5 = 0.$

$\rightarrow$ Minimum VC with cardinality: $X_c = x = 2$

# Cutting Planes (CP)




$x_1 = x_2 = x_3 = 0.5 \quad x_1 = 0, x_2 = x_3 = 1$

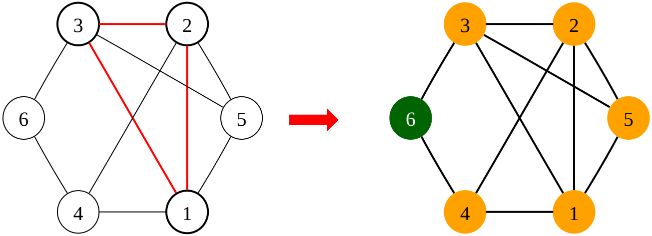
**Idea:** Limit solution space by adding extra constraints (CPs)

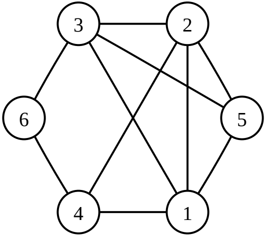
**Loops:**

- Search random loop of length $l$ (spanning tree + edge)
- Add constraint (CP) to LP: $$\sum_{i \in \text{loop}} x_i \geq \left\lceil \frac{l}{2} \right\rceil, \quad (*)$$ if loop has odd length $l$ and $(*)$ is not fulfilled yet.
- Solve LP again
- Extensions: subgraphs; branch & cut

# Example for CP approach

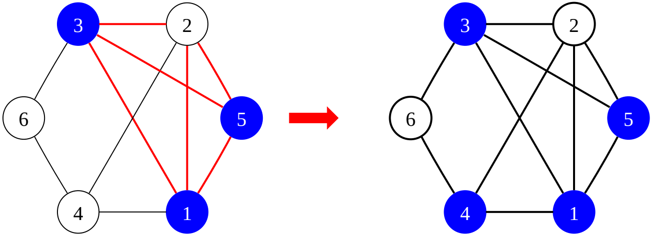
**Loops:**



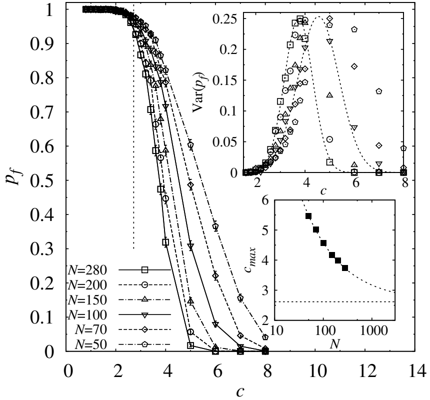

$x_i = 2/3$ $x_i = 1/3$ $x_i = 1$

**Subgraphs:**



# Results cutting planes

Fraction $p_f$ of completely solved solutions.




[T. Dewenter & AKH, preprint arXiv 2012] (old version)

Peak of fluctuations (top inset):
Position converges to $c_{\text{crit}} = 2.6(1)$ (bottom inset)
compatible with $c = e$.

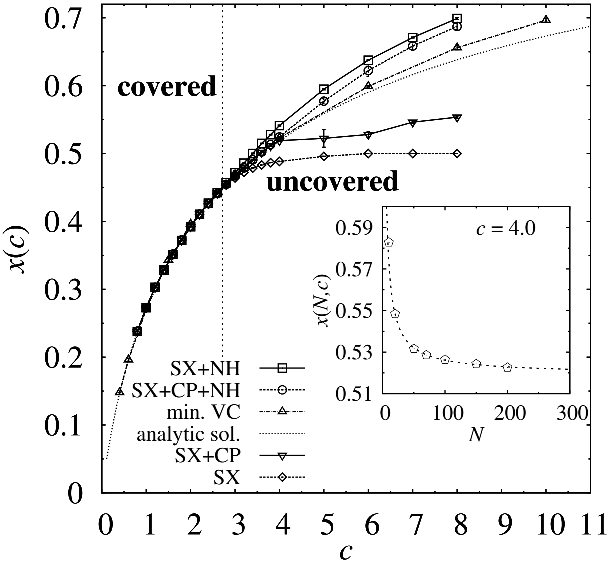
# LP phase diagram

SX+CP:
lower bounds

NH:
"node heuristics"
set some $x_i = 0$
and repeat LP
$\rightarrow$ true VC
$\rightarrow$ upper bounds

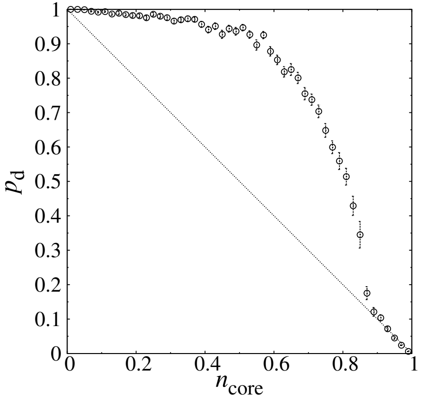
upper $\approx$ lower
bound for $c \leq e$

several sizes $N$
finite-size scaling:
$x(N) = x_\infty + aN^{-b}$



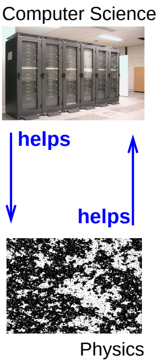
# Correspondence to core

Hard instances (leaf removal & branch-and-cut algorithm)
$\leftrightarrow$ hard instances (LP & cutting planes) ?
$p_{\mathrm{d}}$: fraction of variables $\in \{0, 1\}$, $n_{\mathrm{core}}$: relative core size




For ALL instances $p_{\mathrm{d}} \geq n_{\mathrm{core}}$ !

# Summary




- Simple yet complex-behaving model: Vertex-cover problem
- Complexity Theory: NP-complete
- Simple/medium complex algorithms:
  Heuristisc
  Leaf removal
  Branch-and-bound algorithm
  Linear Programming & cutting planes
- Physics: phase-transition in solvability/running time